B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

P MANYA

(1BM22CS187)

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi,

Bangalore, 560 019

2023-2024.

INDEX

Sl.No.	Title	Date
1	Complete scanned Observation Book	12/12/2023 - 20/02/2024
2	Lab 1	12/12/2023
3	Lab 2	19/12/2023
4	Lab 3	26/12/2023
5	Lab 4	02/01/2024
6	Lab 5	09/01/2024
7	Lab 6	16/01/2024
8	Lab 7	23/01/2024
9	Lab 8	30/01/2024
10	Lab 9	06/02/2024
11	Lab 10	20/02/2024

Dt:

Date __/__/__

Page

1.
```
class RectangleArea {
public static void main (String args[]) {
int length; breadth;
length = Integer.parseInt (args[0]);
breadth = Integer.parseInt (args[1]);
int area = length * breadth;
System.out.println (" length of rectangle= "
                    + length);
System.out.println (" breadth of rectangle="
                    + breadth);
System.out.println (" area is = " + area).
}}
```

SCANNER

2.
```
import java.util.Scanner;
class HelloWorld {
public static void main (String args[])
{
int a, float b; String s;
Scanner in = new Scanner (System.in);
System.out.println (" Enter a string ").
s = in.nextline();
System.out.println (" You entered string "+s).
System.out.println (" Enter an integer ").
a = in.nextInt();
System.out.println (" You enter int "+a);
System.out.println (" Enter float ");
b = in.nextFloat();
System.out.println (" You entered float "+b).
}}
```

3. ARRAY

```
class AutoArray {
public static void main (String[] args[])
int month_days[] = {31, 28, 31, 30, 31, 30,
31, 31, 30, 31, 30, 31};
System.out.println("April has" + month_days[3] + "days")
}}
```

4. FACTORIAL

```
class factorial {
public static void main (String args[])
{
 int fac = 1;
System.out.println("Enter no");
Scanner sc = new Scanner (System.in);
int n = sc.nextInt();
for (int i = 1; i <= n; i++) {
   fac = fac * i;
}
System.out.println("The fact: " + fac)
}}
```

Date __/__/__

Page _____

5. PALINDROME

```
class palindrome {
public static void main (String args[])
{
int n, t, rem, rev=0;
Scanner sc= new Scanner (System.in);
System.out.println(" Enter 5-digits");
n = sc.nextInt();
t= n;
while (t>0) {
rem = t%10;
rev= rev * 10+ rem;
t= t/10;
}
if (rev ==n) {
System.out.println(" Palindrome ");
}
else {
System.out.println(" not palindrome");
} } }
```

6.

```
class sum of digits {
public static void main (String args[]) {
long number, sum;
Scanner sc= new Scanner (System.in);
System.out.println(" Enter 5digit no");
number = sc.nextLong();
for (sum= 0; number!=0; number number/10){
sum = sum + number%10;
}
System.out.println(" Sum of digits "+sum);
} }
```

Date __/__/__
Page ____

7. class conversion {
public static void main (String args[])
{

```
byte b=1; short s1=1000, s2;
int i1 =100000, i2;
long l1=1000000, l2;
char c='+';
float f1= 25.69f, f2;
double d1= 536987.125, d2;
System.out.println("b+" "+s1+"t""+i1+"t""
  l1+"t" "+c+"t""+f1+"t"+d1);
s2 =b;
i2 = s1;
l2 =i1;
System.out.println("s2""t"+i2+"t""+l2)
}
```

Date __/__/__

Page _____

Quadratic Eq:

```
import java.util.Scanner;
class Quadratic
{
int a, b, c;
double r1, r2, d;
void get d ()
{
Scanner s = new Scanner (System.in);
System.out.println("Enter coefficients of a, b, c");
a = s.nextInt ();
b = s.nextInt ();
c = s.nextInt();
}
void compute()
{
while (a==0)
{
System.out.println("Not a quad Eq");
System.out.println("Enter non-zero value for a");
Scanner s = new Scanner (System.in);
a = s.nextInt();
}
d = b*b-4*a*c;
if (d==0)
{
r1 = (-b)/(2*a);
System.out.println("Roots are real & equal")
System.out.println("Root1 = Root2 = "+r1);
}
```

Date __/__/__
Page

```
else if (d>0)
{
r1 = ((-b) + (Math.sqrt(d)))/(double)(2*a);
r2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);
System.out.println("Roots are real & distinct
System.out.println("Root1 =" + r1 + "Root2=" +
}
else if (d<0)
{
System.out.println("Roots are imaginary
r1 = (-b)/(2*a);
r2 = Math.sqrt(-d)/(2*a);
System.out.println("Root1=" + r1 + "+i" + r2)
System.out.println("Root1=" + r1 + "-i" + r2)
}
}}
class QuadraticMain
{
public static void main(String args[])
{
Quadratic q = new Quadratic();
q.getd();
q.compute();
}}
```

Output

Enter coefficients of a,b,c
2 6 2
Roots are real & distinct
Root1 = -0.38196 Root2 = -2.6180

Date __/__/__

Page _____

12/ 12/ 12/

Roots are unimaginary

Root 1 = 0.0 + i 0.8660

Root 1 = 0.0 - i 0.8660

0 0 0

Not a quadratic equation

Enter a non-zero value for a

1 0 0

Roots are real and equal

Root = Root 2 = 0.0

Rm 12/12/23

SGPA Calculation

```
import java.util.Scanner;
class Subject {
int subjectMarks, credits, grade; }
class student {
  string name;
  string usn;
  double SGPA;
  Scanner s;
  Subject subjects[];
student()
 {
  int i;
  subjects = new Subject[9];
  for (i=0; i<8; i++)
  subjects [i] = new subject();
  s = new Scanner (System.in);
  }
  public void getStudentDetails(){
System.out.println ("Enter stu name
name = s.nextline();
System.out.println ("Enter student us
usn = s.nextline();
public void getMarks () {
  int i;
  for (i=0; i<8; i++) {
System.out.println ("Enter marks " + (i+1) +
Subject [i].subjectMarks = s.next Int()
if (Subjects [i].subject Marks >=40 &&
Subject [i].subjects Marks <=100) {
Subjects [i].grade = calculate Grade (Subjects
[i].subject Marks);
else {
```

```
System.out.println("Invalid");
System.out.println("enter credits");
Subjects[i].credits = s.nextInt();
}}
public int calculationGrade(int marks)
{
if(marks >=90)
return 10;
else if (marks >= 70 && marks <= 80)
return9;
else if (marks >= 60 && marks <= 70)
return8;
else if (marks >= 50 && marks <= 60)
return7;
else
return6;
}
public void compute SGPA() {
int total score=0;
int total cred=0;
for(int i=0; i<5; i++){
total score += Subjects[i].grade *
Subjects[i].credits;
total cred += Subjects[i].credits;
}
SGPA = (double) totalscore / (double)
totalcred;
}}
public class stud {
public static void main (String arg[])
{
student s1 = new student();
s1.getstudentDetails();
```

```
s1.setMarks();
s1.computeSGPA();
System.out.println("St name:"+s1.name);
System.out.println("student USN"+s1.usn);
System.out.println("SGPA:"+s1.SGPA);
}
```

Output

```
Enter student name
poojam
student USN
1BM22CS001
Marks of sub 1
7
Invalid
enter credits
Enter sub 2
78
enter credits
3
enter sub 3
100
enter credits
3
Enter marks of sub 4
Invalid
enter credits
4
enter marks of sub 5
79
enter credits
```

Date __/__/__

Page _____

enter marks of sub 6
78
enter credits
3
enter marks of sub 7
40
enter credits
3
Enter marks of sub 8
100
enter credits
4
Student name pooja
USN - 1BM20CS001
SGPA = 6.23

Pr
18/12/2023

Date __/__/__
Page ____

26/12/23 LAB-3, Details of books

```
import java.util.Scanner;
class book {
    private String name;
    private string name;
    private double price;
    private int num Pages;
public book (string name, string author
double price, int num Pages) {
this.name = name;
this.author = author;
this.price = price;
this.num pages = num pages;
}

public void set name (string name) {
  this.name = name;
}

  public string getName () {
    return name;
}

  public void set author (string
                          author) {
  this.author = author;
}

  public string get author () {
      return author;
  }

  public void set price (double price) {
      this.price = price;
  }

  public double get price () {
      return price }
```

Date __/__/__

Page _____

```
public void set numPages() {
        return numPages;
    }
    public int get numPages() {
        return numPages;
    }
    public string to String () {
        return " Book Details :\nName :" + name +
"\n Author : " + author + "\n Price : INR " + price
+ "\n Number of pages : " + numPages;
    }}
public class Main {
    public static void main (String[] args){
Scanner scanner = new Scanner(System.in);
System.out.println(" Enter no. of books ");
    int n = scanner.next int ();
    Book[] books = new Book[n];
for (int i=0; i<n; i++){
System.out.println (" Enter details " +(i+1)+ :");
scanner.next line ();
System.out.println (" Enter name ");
String name = scanner.next line();
System.out.println(" Author ");
String author = scanner.next line();
System.out.println (" Price ");
double price = scanner.next line();
System.out.println(" No. of pages ");
int numPages = scanner.next line();
book[i] = new Book(name, author, price,
numpages);
System.out.println(" Details is ");
for (int i=0; i<n; i++){
System.out.println("\nBook" + (i+1) + ":\n" + book
    }
```

scanner.close()
}}

Output:

Enter no. of books
2
Enter details
Enter name
The world
Author:
Km jot
Price:
635.78
No. of pages
300
Enter details
Enter name
The space & stars
Author
Samhita
Price
353.53
No. of pages
200
Details of
Book 1
Name: The world
Author: Km jot
Price: 635.78
No. of pages: 300

Date __/__/__

Page _____

Book 2

Name: Spencer Stars

Author: d m Lamhung

Price: 353.53

No. of pages: 268

Br
21/12/2024

Date __/__/__
Page

2/1/24 LAB-4 AREAS OF TRIANGLES, RECTANGLE AND CIRCLE

```
import java.util.Scanner;
abstract class shape {
    protected int sidea;
    protected int sideb;
public shape (int sidea, int sideb
    this.sidea = sidea;
    this.sideb = sideb;
    }
    public abstract void printArea();
}
class Rectangle extends shape {
  public Rectangle (int length, int width
  super (length, width);
}
  public void printArea() {
   int area = sidea * sideb;
  System.out.println(" Area of Rectangle:
                                  area);
  }}
class Triangle extends shape {
  public Triangle (int base, int height
  super (base, height);
  }
  public void printArea() {
   double area = 0.5 * sidea * sideb;
  System.out.println(" Area of Triangle
  }}                                  + area);
class Circle extends shape {
  public Circle (int radius) {
    super (radius, 0); }
```

Date __/__/__

Page _____

```
class circle {: public void printArea () {
double area = MATH.PI * side a * side b;
System.out.println("Area of Circle:"
                    + area);
}}
public class shapes {
  public static void main(String [] args) {
  Scanner scanner = new Scanner(System
                                .in);
System.out.println("Enter length & width
                    of Rectangle:");
int rectlength = scanner.nextInt();
int rectwidth = scanner.nextInt();
Triangle triangle = new Rectangle(rectlength,
                    rect width);
System.out.println("Enter h & b for tri");
int tribase = scanner.nextInt();
int rectwidth = scanner.nextInt();
Triangle triangle = new Triangle(tribase,
                    tri height);
System.out.print("Enter radius:");
  int circleRadius = scanner.nextInt();
Circle circle = new circle(circleRadius);
scanner.close();
rectangle.printArea();
triangle.printArea();
circle.printArea();
}
}
```

Output:

Enter length & breadth for rectangle: 2
3

Enter base and height for Triangle: 5
4

Enter radius for circle : 9
Area of Rectangle : 6
Area of Triangle : 10.0
Area of Circle : 0.0 check it 254.46

Date __/__/__

Page _____

9/1/24 LAB PROGRAM - 5 Bank Account

```
import java.util.Scanner;
class Account {
    String customerName;
    long accountNumber;
    String accountType;
    double balance;

    public Account(String customerName,
        long accountNumber, String accountType,
        double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of $" +
        amount + " successful. Updated balance
        : $" + balance);
    }
    public void displayBalance() {
        System.out.println("Acc Balance:
        $" + balance);
    }
}
class CurAccount extends Account {
    double minBalance;
    double serviceCharge;
```

Date __/__/__
Page _____

```
public CurrentAccount (String customer
    Name, long accountNumber,
    double balance, double balance,
    double minBalance, double service
    charge) {
        super(customerName, account
            Number, "Current", balance);
        this.minBalance = minBalance;
        this.serviceCharge = serviceCharge;
    }

    public void checkMinBalance() {
        if(balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Min bal not
            maintained service charge of $" +
            serviceCharge + " imposed");
            displayBalance();
        }
    }

    public void withdraw(double amount) {
        if(amount > balance) {
            System.out.println("Insufficient
            funds. Withdrawal failed");
        }
        else {
            balance -= amount;
            System.out.println("Withdrawal of
            $" + amount + " successful. Updated
            balance: $" + balance);
            checkMinBalance();
        }
    }
}
```

Date __/__/__

Page _____

```
class SavAccount extends Account {
    double interestRate;

    public SavAccount(String customerName,
long accountNumber, double balance,
double interestRate) {
        super(customerName, accountNumber,
"Savings", balance);
        this.interestRate = interestRate;
    }
    public void computeInterest() {
        double interest = balance * (interestRate
/100);
        balance += interest;
        System.out.println("Interest computed
and deposited: $" + interest);
        displayBalance();
    }

    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient funds.
Withdrawal failed.");
        }
        else {
            balance -= amount;
            System.out.println("Withdrawal of $"
+ amount + " successful. Updated balance:
$" + balance);
        }
    }
}
public class Bank {
    public static void main(String[] a[]) {
```

Date __/__/__
Page _____

```
Scanner scanner = new Scanner(System.in);

CurAccount currentAccount = new CurAccount
("J", 12345678 90, 1000, 800, 10);
SavAccount savingsAccount = new
SavAccount ("D", 98452314, 2000, 5);

int choice;
do {
    System.out.println("Select an option:");
    System.out.println("1. Deposit");
    System.out.println("2. Disp. balance");
    System.out.println("3. Compute Interest");
    System.out.println("4. Withdraw");
    System.out.println("5. Exit");
    System.out.println("Enter your choice");
    choice = scanner.nextInt();

switch(choice) {
Case 1:
    System.out.println("Enter amount
                to deposit");
    double depositAmount = scanner.
                nextDouble();
System.out.println("Select account(1.
    Current, 2. Savings):");
int accType = scanner.nextInt();
if (accType == 1) {
    currentAccount.displayBalance();
}
else if (accType == 2)
{
    savingsAccount.displayBalance();
```

Date __/__/__

Page _____

```
}
else
{
    System.out.println("Invalid acc type");
}
 break;
case 2:
 System.out.println("Select acc (1. Current,
   2. Savings):");
    int accType = 1)
{
        CurrentAccount.displayBalance();
}
 else if (accType == 2)
{
 SavingsAccount.displayBalance();
}
 else {
 System.out.println("Invalid");
}
 break;
Case 3:
 if (SavingAccount instance of
      SavAccount)
 {
 ((SavAccount) SavingsAccount).computeInterest();
}
 else {
   System.out.println("Invalid");
}
 break;
Case 4;
```

Date ___/___/___
Page ___

```
System.out.println("Enter amount to be
withdrawn");
    double withdrawAmount = scanner.
nextDouble();
    System.out.println("Select Acc (1. Current
2. Savings):");
        int accTyp = scanner.nextInt();
    if (accTyp == 1)
{
    currentAccount.withdraw (with
                drawAmt);
}
    else if (acc Typ == 2)
{
    SavingsAccount.withdraw (withdraw
                Acc);
}
    else {
    System.out.println("Invalid");
}
    break;
    case 5:
    System.out.println("Exiting");
break;
    default:
    System.out.println("Invalid");
}
}
while (choice != 5);
scanner.close();
}
```

Date __/__/__

Page _____

Output

Select an option
1. Deposit
2. Display Balance
3. Compute Interest
4. Withdraw
5. Exit
Enter your choice : 1
Enter amt to deposit: 1000
Select account (1. current 2. Savings): 2
Deposit of $10000.0 successful.

Select option:
1. Deposit
2. Display Balance:
3. Compute Balance
4. Withdraw
5. Exit
Enter your choice: 2
Select acc (1. 2.) 1
Account balance $1000.0

Select an option
1. Deposit
2. Display balance
3. Compute Balance
4. Withdraw
5. Exit
Enter your choice: 3
Interest computed & deposited: $600.0
Acct balance: $12600.0

Date __/__/__
Page ____

Select an option
1. Deposit
2. Display balance
3. Compute balance
4. Exit
Enter your choice: 4
Enter amt to withdraw: 5000
Select account (1. Current, 2. Savings): 1
Insufficient

Select an option
1. Deposit
2. Display balance
3. Compute balance
4. Exit
5. Enter your choice: 5
Exiting.

9/1/2024

Date __/__/__

Page _____

16/1/24 PRACTISE PROGRAMS:

1. Hello, World!
Hello
Java Programming
Java
ABCDE
Java
BCD

2. String 1: Hello
String 2: Java
Concatenated String: Hello, Java!
Length of Concatenated string: 12

3,4. College name = 'Welcome to DMSCE College'
Extracted Substring: Dmsce

5. Bytes Representation
72 101 108 108 111 44 32 87 111 114
108 100 33
Original string: Hello, World!
char Array representation:
J a v a P r o g r a m m i n g
Original string: Java Programming

6. Dmsce equals Dmsce -> true
Dmsce equals College -> false
Dmsce equals DMSCE -> false
Dmsce equals ignore case DMSCE -> true

Date __/__/__
Page ____

7. Substring is matched

8.9. Hello, world! starts with Hello -> true
Java Programming starts with Python
Hello, world! ends with world!
Java Programming ends with C++

10. Hello! equals Hello! -> true
Hello! == Hello! -> false

12. Sorted Numbers (Descending):
10
9
8
7
6
5
4
3
2
1

13. Concatenated string: helloworld

14. Modified string: This is a test.
This is, too.

15. Original string: Hello, strings!
Trimmed string: Hello, Friends!

16. Original: This is my college
Modified: This is my commerce.

Date __/__/__

Page _____

16. Enter details of student 1:
Reg Number: 1
Full Name: A
Sem: 1
CGPA: 9

Enter details of student 2:
Reg Number: 2
Full Name: B
Sem: 2
CGPA: 8.1

18. After setLength(5): Hello
character at index: 1: e
After setCharAt(1, 'a'): Hallo
characters from index 0 to 4: Hallo
After append(): Hallo Appended!
After insert(7, 'Inserted'): Hallo
All Inserted Appended!
After reverse(): ! dedneppA dellerI olla H
After delete(5, 14): ! dednsnI A olla H
After deleteCharAt(0): dednsnI A olla H
After replace(0, 4, 'Replaced'): Replaced nsnI
olla H
Substring from index 3 to 7: laced

19. Details about Eagle
Eagle flies high with powerful wings

Details about Hawks.
Hawk emits a high pitched screech
while flying.

Date ___/___/___
Page ___

20. Details about circle:
Area : 78.5398
Perimeter : 31.4159

Details about Triangle :
Area : 6.0
Perimeter : 12.0

11. apple
ball
cat
dog
ENT
free
gun
hi
hello
hand
hello
parrot
green
drink
star
tree
umbrella
rain
watch
xmas
yatch

Rm 16/11/2024

Date __/__/__
Page _____

23/1/24 LAB PROGRAM-6

PACKAGES:

```
package CIE;
import java.util.Scanner;
public class student
{
    int USN;
    String Name;
    int Sem;
    Scanner sc = new Scanner(System.in);
    public void accept()
    {
        System.out.println("USN");
        USN = sc.nextInt();
        System.out.println("Name");
        Name = sc.next();
        System.out.println("Sem");
        Sem = sc.nextInt();
    }
    public void display()
    {
        System.out.println("Name is:"+Name);
        System.out.println("USN:"+USN);
        System.out.println("Sem"+Sem);
    }
}
```

```
package CIE;
import java.util.Scanner;
public class Internals Extends Student
{
    public int IMarks[] = new int[5];
    public void accept()
{
    System.out.println("Enter marks in
    Scanner.out.~~println(s)~~ = new
    Scanner(System.in);
for(int i=0; i<5; i++)
{
    System.out.println("Marks in
+(i+1)+" subject is ");
IMarks[i]=(Sc.nextInt());
}
}
public void display()
{
    System.out.println("Internal
for(int i=0; i<5; i++)
{
System.out.println("Marks in
+(i+1)+" subject is " + IMarks
[i]);
}
}
}
```

Date __/__/__
Page _____

```
package SEE;
import CIE.Student;
import java.util.Scanner;
public class Externals extends Student
{
    public int EMarks[]=new int[5];
    public void accept()
    {
        System.out.println("Enter marks in
                5 subjects");
        Scanner S1=new Scanner(System.in);
        for(int i=0; i<5; i++)
        {
            System.out.println("Marks"+(i+1)+"
                    Subject:");
            EMarks[i]=S1.nextInt();
        }
    }
    public void display()
    {
        System.out.println("External Details")
        for(int i=0; i<5; i++)
        {
            System.out.println("Marks in:"+(i+1)
                    +" Subject:"+EMarks[i]);
        }
    }
}
```

Date __/__/__

Page _____

```
import CIE.Student;
import CIE.Internals;
import SEE.Externals;
import java.util.Scanner;
public class Final
{
    public static void main(String[])
{
    Scanner S1= new Scanner(System.in);
int n;
System.out.println("Enter an value");
n=S1.nextInt();
CIE.Student S[]= new CIE.Student[n];
CIE.Internals I[]= new CIE.Internals[n];
SEE.Externals E[]= new SEE.Externals[n];
for(int i=0;i<n;i++)
{
    S[i]= new CIE.Student();
    I[i]= new CIE.Internals();
    E[i]= new SEE.Externals();
    S[i].accept();
    I[i].accept();
    E[i].accept();
    S[i].display();
    I[i].display();
    E[i].display();
for(int j=0;j<5;j++)
{
int finalmarks = I[i].IMarks[j]+
                 E[i].EMarks[j];
System.out.println("Marks in "+(i+1)+"
    Subjects is "+final marks);
}
```

Date __/__/__
Page _____

output
Enter n value
1
USN
1
Name
A
Sem
I
Enter Marks in 5 subjects
45
34
50
44
46
Enter SEE marks in 5 subjects
33
45
50
44
45
Name is : A
USN is 1
Sem is : 1
Internal Details
Marks in 1 subject : 45
Marks in 2 subject : 34
Marks in 3 subject : 50
Marks in 4 subject : 44

Marks in 5 subject : 46

External Details:
Marks in 1 subject = 33
Marks in 2 subject : 45
Marks in 3 subject : 50
Marks in 4 subject : 44
Marks in 5 subject : 45

Total Marks:
Marks in 1 subject : 78
Marks in 2 subject : 79
Marks in 3 subject : 100
Marks in 4 subject : 88
Marks in 5 subject : 91

28/1/24

Date __/__/__
Page _____

Week-7 · Exception Handling

```
import java.util.Scanner;
class WrongAge extends Exception
{
  public WrongAge (String message)
{
    super( message);
}
}

  class Father
{
    protected int fatherAge;
  public Father( int age) throws
  WrongAge
{
    fatherAge = age;
  if (fatherAge<0)
{
    throw new WrongAge("Fathers age
                  can't be negative");
  }
}
}

class son extends Father
{
    private int sonAge;
  public Son (int FatherAge, int SonAge)
throws WrongAge
{
    super( fatherAge);
    this.sonAge = SonAge;
```

```
if (sonAge <= 0)
{
    throw new ("Son's age can't be neg");
}
if (sonAge >= fatherAge)
{
    throw new WrongAge("Son's age can't
            be negative >= father's age");
    }
}
}

public class Main {
    public static void main (String [] ar
    {
        Scanner scanner = new Scanner
                (System.in);

try {
    System.out.println("Enter father's
                age");
    int fatherAge = scanner.nextInt();
    System.out.println("Enter Son's Age")
    int sonAge = scanner.nextInt();

    Son son = new Son (fatherAge, sonAg

    System.out.println("Father's Age"+
                fatherAge);
    System.out.println("Son's age"+
                sonAge);
}
catch (WrongAge e)
```

Date __/__/__
Page _____

```
{
System.out.println("Exception caught" +e);

System.out.println("Exception caught" +e.getMessage());

}
catch(Exception e)
{
System.out.println("Error" +e);
System.out.println("Error" +e.getMessage());
}
finally {
Scanner.close();
}
}
}
```

Output:
Enter father's age : 50
Enter Son's age : 20
Exception caught : Wrong Age! Son's age can't be neg or zero.
Exception caught: Son's age can't be neg or zero.

Enter father's age : 30
Enter son age : 31
Exception caught: Son's age can't be greater or equal to father's age.

30/1/2024

6/2/24 LAB-8 : Threads:

```
class displayThread extends Thread {
    private String message;
    private int interval;
    private boolean running = true;
public displayThread (String message,
                      int interval)
{
    this.message = message;
    this.interval = interval;
}
    public void run()
    {
      while (running)
      {
        System.out.println(message);
      try
      {
        Thread.sleep (interval);
      }
      catch (InterruptedException e)
      {
        e.printStackTrace();
      }
    }
}

    public void stopThread()
    {
      running = false;
    }
}

public class Thread1 {
```

```
public static void main (String [] args)
{
    DisplayThread bmsThread = new DisplayThread
            ("BMS College of Engineering", 10000);
    DisplayThread cseThread = new DisplayThread
            ("CSE", 2000);

    bmsThread.start();
    cseThread.start();
    System.out.println("Presenter to stop");
    try
    {
        System.in.read();
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
    bmsThread.stopThread();
    cseThread.stopThread();
    }
}
```

Output:

BMS College of Engineering
CSE
CSE
CSE
CSE
CSE

6/2/24 LAB: 10 IPC

```
class Q {
int n;
boolean value set = false;
synchronized int get()
{
while (! value set)
try
{
wait();
}
catch (InterruptedException e)
{
System.out.println("Got:" + n);
value set = false;
notify();
return n;
}
synchronized void put(int n)
{
while (value set)
try
{
wait();
}
catch (InterruptedException e)
{
System.out.println("Interrupted");
}
this.n = n;
value set = true;
System.out.println("Put:" + n);
```

Date ___/___/___
Page _____

```
notify();
}
}
class Producer implements Runnable {
Q q;
Producer (Q q)
{
this.q=q;
new Thread (this, "Producer").start();
}
public void run()
{
int i=0;
while (i<15)
{
q.put(i++);
}
}
}
class Consumer implements Runnable
{
Q q;
Consumer( Q q)
{
this.q=q;
new Thread(this, "Consumer").start()
}
public void run()
{
int i=0;
while (i<15)
{
```

```
int x = q.get();
  i++;
  }
}
}

class PCFixed {
public static void main () {
  Q q = new Q();
  new Producer(q);
  new Consumer(q);
  System.out.println("C to stop");
}
}
```

Output:

Put: 0	Put: 8
Got: 0	Got: 8
Put: 1	Put: 9
Got: 1	Got: 9
Put: 2	Put: 10
Got: 2	Got: 10
Put: 3	Put: 11
Got: 3	Got: 11
Put: 4	Put: 12
Got: 4	Got: 12
Put: 5	Put: 13
Got: 5	Got: 13
Put: 6	Put: 14
Got: 6	Got: 14
Put: 7	
Got: 7	

Date __/__/__
Page _____

Deadlock

13/2/24

```
class A
{
synchronised void foo(B b)
{
  String name = Thread.currentThread().
          getName();
System.out.println(name + " entered A.
                foo");
try {
Thread.sleep(1000);
}
catch (Exception e)
{
System.out.println(" A interrupted");
}
System.out.println(name + " trying
to call B.last()");
  b.last();
}
  void last() {
   System.out.println(" Inside A.last");
  }
}
class B {
Synchronised void bar(A a) {
String name = Thread.currentThread().
          getName();
System.out.println(name + " entered B.bar");
try {
  Thread.sleep(1000);
}
```

```
catch (Exception e)
  {
  System.out.println("B exit");
  }
  System.out.println("Name + "trying to
                call A.last()");
  a.last();
  }
  void last()
  {
  System.out.println("Inside A.last");
  }
  }
class Deadlock implements Runnable
{
A a = new A();
B b = new B();
Deadlock()
{
Thread.currentThread().setName("
                Main Thread");
Thread t = new Thread(this,"Racing Thread
t.start();
a.foo(b);
System.out.println("Back in main
                thread");
}
public void run()
{
b.bar(a);
System.out.println("Back in other
                thread");
```

Date __/__/__
Page _____

```
public static void main(String args[]){
    new Deadlock();
    }
}
```

output:

```
Main Thread entered A.foo()
Racing Thread trying to call B.last()
Inside A.last method
back in main thread
Racing Thread trying to call A.last()
Inside A.last
Lock released by Thread-0
```

13/2/2024

20/2/24 LAB: 9 Integer Division.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
SwingDemo () {
    JFrame jfrm = new JFrame("Divider App");
    jfrm.setSize(275, 150);
    jfrm.setLayout(new FlowLayout());
    jfrm.setDefaultCloseOperation(JFrame.
                EXIT_ON_CLOSE);
JLabel jlab = new JLabel("Enter the
dividend and divider");

JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);

JButton button = new JButton("Calculate

JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();

jfrm.add(err);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
```

Date __/__/__
Page _____

```
Action listner l = new Action listner() {
  public void actionPerformed(Action
                              event e) {
System.out.println("Action event
                   from a text field");
  }
};

ajtf.addActionListner(l);
bjtf.addActionListner(l);

button.addActionListner(new Action
                        listner() {
public void actionPerformed(Action
                            Event e)
{
try {
  int a = Integer.parseInt(ajtf.get
                           Text());
int b = Integer.parseInt(bjtf.getText());
int ans = a/b;
alab.setText("\n A=" + a);
blab.setText("\n B=" + b);
anslab.setText("\n Ans=" + ans);
}
catch (NumberFormatException e) {
  alab.setText(" ");
  blab.setText(" ");
  anslab.setText(" ");
  err.setText(" Enter only Integers!");
}
```

```
}
catch(ArithmaticException e)
{
    alab.setText(" ");
    blab.setText(" ");
    anslab.setText(" ");
    err.setText(" Enter only Int ");
}
    catch (ArithmaticException e)
    {
      alab.setText(" ");
      blab.setText(" ");
      anslab.setText(" ");
      err.setText(" B should be Non zero ");
    }
}
}
frm.setVisible(true);
}
public static void main(String args[])
{
String Vthis=this.invokeLater(new
                Runnable() {
public void run() {
new String Demo();
}});
}
}
```

Date __/__/__
Page _____

Output

Enter divident divisor
100 50

[Calculate] A = 100 B = 50 Ans: 2

AWT Functions

1. JFrame - class in java that is used for creating graphical user interfaces in java application.

2. SetSize() - is a method which is used with components such as Jpanel. to set size.

3. SetLayout() - used to set layout manager for a container such as JFrame or any other container component.

4. SetDefaultCloseOp() - used to specify default close operator for a JFrame

5. JLabel - used to display non editable text or image.

6. JTextField - provides a text input field in GUI.

7. AddActionListener - used to register an ActionListener for a button or any other interactive component that generates

Date __/__/__

Page ____

8- Set Text: It is a method used to content of text based components.

restrict text into rules

02 001

s: int 02: 2 001: A [statement]

Rw 20/2/24

Assignment TCIA

1. Transmission - Short range devices provide wireless connectivity

2. Digital Journey - Digitalize throughout value chain

3. Digital at Heart - Digitalize core

4. Digital at Scale - Transformation revenues and operation transformation revenues scale from

5. Digital at Speed - Transformation [illegible]

Lab 1

Develop a Java program that prints all real solutions to the quadratic equation ax 2 +bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b^2 – 4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner; class
Quadratic
{ int a, b, c;
double r1, r2, d;
void getd()
{
Scanner s = new Scanner(System.in);
System.out.println("Enter the coefficients of a,b,c");
a = s.nextInt(); b = s.nextInt(); c = s.nextInt();
} void
compute()
{ while(a==0)
{
System.out.println("Not a quadratic equation");
System.out.println("Enter a non zero value for a:");
Scanner s = new Scanner(System.in); a =
s.nextInt();
} d = b*b-
4*a*c;
if(d==0)
{ r1 = (-
b)/(2*a);
System.out.println("Roots are real and equal");
System.out.println("Roo1 = Root2 = " + r1);
```

```
} else if(d&gt;0) { r1 = ((-
b)+(Math.sqrt(d)))/(double)(2*a); r2 = ((-b)-
(Math.sqrt(d)))/(double)(2*a);
System.out.println("Roots are real and
distinct");
System.out.println("Roo1 = " + r1 + " Root2 = " + r2);
} else
if(d&lt;0)
{
System.out.println("Roots are imaginary");
r1 = (-b)/(2*a); r2 = Math.sqrt(-d)/(2*a);
System.out.println("Root1 = " + r1 + " + i"+r2);
System.out.println("Root1 = " + r1 + " - i"+r2);
}
} } class
QuadraticMain
{ public static void main(String
args[])
{
Quadratic q = new Quadratic(); q.getd();
q.compute();
}
}
```

Output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Admin\Desktop\1BM22CS187> javac  QuadraticMain.java
PS C:\Users\Admin\Desktop\1BM22CS187> java  QuadraticMain
Enter the coefficients of a,b,c
2 6 2
Roots are real and distinct
Roo1 = -0.3819660112501051 Root2 = -2.618033988749895
PS C:\Users\Admin\Desktop\1BM22CS187>  javac  QuadraticMain.java
PS C:\Users\Admin\Desktop\1BM22CS187> java  QuadraticMain
Enter the coefficients of a,b,c
121
 121
121
Roots are imaginary
Root1 = 0.0 + i0.8660254037844386
Root1 = 0.0 - i0.8660254037844386
PS C:\Users\Admin\Desktop\1BM22CS187> javac  QuadraticMain.java
PS C:\Users\Admin\Desktop\1BM22CS187> java  QuadraticMain
Enter the coefficients of a,b,c
0
0
0
Not a quadratic equation
Enter a non zero value for a:
1
Roots are real and equal
```

Lab 2

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

import java.util.Scanner; class

subject{ int subjectMarks,

```
credits, grade;} class Student {
String name;
   String usn;
double SGPA;
Scanner s;     subject
subjects[];
Student() {
int i;
subjects = new subject[9];
for(i=0;i<8;i++) subjects[i]
= new subject(); s = new
Scanner(System.in);
}
public void getStudentDetails(){ System.out.println("Enter
student name:"); name=s.nextLine();
System.out.println("Enter Student USN:");
usn=s.nextLine();} public void
getMarks(){
int i;
for(i=0;i<8;i++){
System.out.println("Enter marks of subject"+(i+1)+":");
subjects[i].subjectMarks= s.nextInt();
if(subjects[i].subjectMarks>=40&&subjects[i].subjectMarks<=100)
{ subjects[i].grade=calculateGrade(subjects[i].subjectMarks);} else{
System.out.println("Invalid Marks. Marks should be between 40 and 100");}
System.out.println("enter credits:"); subjects[i].credits=s.nextInt();
}
}
```

```
public int calculateGrade(int marks){
if (marks>=90) return 10; else
if(marks>=70&&marks<=80)
return 9;
else if(marks>=60&&marks<=70)
return 8;
else if(marks>=50&&marks<=60)
return 7;
else return
6;
}
public void computeSGPA() {
        int totalscore = 0;         int totalcred = 0;         for (int i
= 0; i < 8; i++) {             totalscore += subjects[i].grade *
subjects[i].credits;             totalcred += subjects[i].credits;
        }
        SGPA = (double) totalscore / (double) totalcred;
    }
} class
Stud{
public static void main(String args[]){
Student s1=new Student();
s1.getStudentDetails(); s1.getMarks();
s1.computeSGPA();
 System.out.println("Student name:"+s1.name);
System.out.println("Student usn:"+s1.usn); System.out.println("Student
sgpa:"+s1.SGPA);}
}
```

Output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Admin\Desktop\1BM22CS187> javac Stud.main
error: Class names, 'Stud.main', are only accepted if annotation processing is explicitly requested
1 error
PS C:\Users\Admin\Desktop\1BM22CS187> javac Stud.java
PS C:\Users\Admin\Desktop\1BM22CS187> java Stud
Enter student name:
pooja
Enter Student USN:
1bm22cs195
Enter marks of subject1:
7
Invalid Marks. Marks should be between 40 and 100
enter credits:
4
Enter marks of subject2:
78
enter credits:
3
Enter marks of subject3:
100
enter credits:
3
Enter marks of subject4:
5
Invalid Marks. Marks should be between 40 and 100
enter credits:
4
Enter marks of subject5:
99
enter credits:
2
Enter marks of subject6:
78
enter credits:
3
Enter marks of subject7:
40
enter credits:
3
Enter marks of subject8:
100
enter credits:
4
Student name:pooja
Student usn:1bm22cs195
Student sgpa:6.230769230769231
```

Lab 3

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;

class Book {     private
```

```
String name;     private
String author;     private
double price;     private
int numPages;     public
Book(String name,
String author, double
price, int numPages) {
this.name = name;
this.author = author;
this.price = price;
this.numPages =
numPages;
   }
   public void setName(String name) {
this.name = name;
   }
   public String getName() {
return name;
   }
   public void setAuthor(String author) {
this.author = author;
   }
   public String getAuthor() {
return author;
   }
   public void setPrice(double price) {
this.price = price;
   }
   public double getPrice() {
return price;
```

```
    }
    public void setNumPages(int numPages) {
this.numPages = numPages;
    }
    public int getNumPages() {
return numPages;

    }
    public String toString() {         return "Book Details: \nName: " + name + "\nAuthor: " +
author + "\nPrice: INR" + price + "\nNumber of Pages: " + numPages;
    }
} public class Main
{
    public static void main(String[] args) {         Scanner
scanner = new Scanner(System.in);
System.out.println("Enter the number of books: ");
int n = scanner.nextInt();         Book[] books = new
Book[n];         for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Book " + (i + 1) + ":");
scanner.nextLine();
            System.out.println("Enter name: ");
            String name = scanner.nextLine();
            System.out.println("Enter author: ");
            String author = scanner.nextLine();
System.out.println("Enter price: ");             double
price = scanner.nextDouble();
            System.out.println("Enter number of pages: ");
int numPages = scanner.nextInt();             books[i] = new
Book(name, author, price, numPages);
        }
```

```
        System.out.println("\nDetails of all books:");
for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":\n" + books[i]);
        }
scanner.close();
    }
}
```

Output:

```
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>cd desktop

C:\Users\admin\Desktop>javac Main.java

C:\Users\admin\Desktop>java Main
Enter the number of books:
2

Enter details for Book 1:
Enter name:
The World
Enter author:
k m joet
Enter price:
635.78
Enter number of pages:
300

Enter details for Book 2:
Enter name:
The Space and stars
Enter author:
d m sanhuy
Enter price:
353.53
Enter number of pages:
268

Details of all books:

Book 1:
Book Details:
Name: The World
Author: k m joet
Price: INR635.78
Number of Pages: 300

Book 2:
Book Details:
Name: The Space and stars
Author: d m sanhuy
Price: INR353.53
Number of Pages: 268
```

Lab 4

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the classShape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

```
import java.util.Scanner; class
InputScanner {
   Scanner s = new Scanner(System.in);
int getInput(String prompt) {
System.out.println(prompt);
      return s.nextInt();
   }
} class shape extends InputScanner
{     double dim1;     double dim2;
shape(double a, double b) {
dim1 = a;        dim2 = b;
   }
}
class Rectangle extends shape {
Rectangle() {        super(0, 0);
```

```
dim1 = getInput("Enter length");
dim2 = getInput("Enter breadth");
   }
   double area() {
      System.out.println("Inside Area for Rectangle.");
return dim1 * dim2;
   }
}
class Triangle extends shape {
Triangle() {         super(0, 0);
dim1 = getInput("Enter length");
dim2 = getInput("Enter base");
   }
   double area() {
      System.out.println("Inside Area for Triangle.");         return dim1 * dim2 / 2;
   } }
class Circle extends shape {     Circle() {
super(0, 0);         dim1 = getInput("Enter
the radius");         dim2 = dim1;
   }
   double area() {
      System.out.println("Inside Area for Circle.");
return Math.PI * dim1 * dim2;
   }
} public class Areas
{
   public static void main(String[] args) {
Rectangle rectangle = new Rectangle();
      System.out.println("Area of Rectangle: " + rectangle.area());
```

```
        Triangle triangle = new Triangle();
        System.out.println("Area of Triangle: " + triangle.area());
        Circle circle = new Circle();
        System.out.println("Area of Circle: " + circle.area());
    }
}
```

Output:

```
C:\Users\bmsce>cd desktop

C:\Users\bmsce\Desktop>javac Shapes.java

C:\Users\bmsce\Desktop>java Shapes
Enter length and width for Rectangle: 2
3
Enter base and height for Triangle: 5
4
Enter radius for Circle: 9
Area of Rectangle: 6
Area of Triangle: 10.0
Area of Circle: 0.0
```

Lab 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
class Account {     String
customerName;     int
accountNumber;
String accountType;
double balance;
   Account(String name, int number, String type, double initialBalance) {
customerName = name;         accountNumber = number;
accountType = type;         balance = initialBalance;
   }
   void deposit(double amount) {
balance += amount;
      System.out.println("Deposit of INR " + amount + " successful");
   }
   void displayBalance() {
      System.out.println("Account Number: " + accountNumber);
      System.out.println("Customer Name: " + customerName);
      System.out.println("Account Type: " + accountType);
      System.out.println("Balance: INR " + balance);
   }
   void withdraw(double amount) {
if (balance >= amount) {
balance -= amount;
         System.out.println("Withdrawal of INR " + amount + " successful");
      } else {
         System.out.println("Insufficient funds");
```

```
        }
    }
    void computeInterest() {
    }
    void checkMinimumBalance(double minBalance, double serviceCharge) {
    }
}
class SavAcct extends Account {
double interestRate = 0.05;
    SavAcct(String name, int number, String type, double initialBalance) {
super(name, number, type, initialBalance);
    }
    void computeInterest() {         double
interest = balance * interestRate;
balance += interest;
        System.out.println("Interest of INR " + interest + " added to the account");
    }
} class CurAcct extends Account
{     double minBalance = 1000;
double serviceCharge = 50;
    CurAcct(String name, int number, String type, double initialBalance) {
super(name, number, type, initialBalance);
    }
    void checkMinimumBalance(double minBalance, double serviceCharge) {
if (balance < minBalance) {
            System.out.println("Service charge of INR " + serviceCharge + " imposed");
balance -= serviceCharge;
        }
    }
```

```
}  public class
Bank {     public
static void
main(String[] args)
{         Scanner
scanner = new
Scanner(System.in
);
System.out.print("
Enter the number
of users: ");
int numUsers =
scanner.nextInt();
        Account[] accounts = new Account[numUsers];
for (int i = 0; i < numUsers; i++) {
System.out.println("\nUser " + (i + 1));
System.out.print("Enter customer name: ");
scanner.nextLine();
            String name = scanner.nextLine();
System.out.print("Enter account number: ");            int
accNumber = scanner.nextInt();
            System.out.print("Enter initial deposit amount: INR ");
double initialDeposit = scanner.nextDouble();
            System.out.print("Enter account type (Savings/Current): ");
scanner.nextLine();
            String accType = scanner.nextLine();            if
(accType.equalsIgnoreCase("Savings")) {                accounts[i] = new
SavAcct(name, accNumber, accType, initialDeposit);
            } else if (accType.equalsIgnoreCase("Current")) {                accounts[i] =
new CurAcct(name, accNumber, accType, initialDeposit);
```

```
        } else {
            System.out.println("Invalid account type entered. Defaulting to Account.");
accounts[i] = new Account(name, accNumber, "Account", initialDeposit);
        }
    }
    boolean exit = false;
while (!exit) {
        System.out.println("\nChoose an option:");
        System.out.println("1. Deposit");
        System.out.println("2. Withdraw");
        System.out.println("3. Display Balance");
        System.out.println("4. Compute Interest (Savings only)");
        System.out.println("5. Exit");
System.out.print("Enter your choice: ");
int choice = scanner.nextInt();            switch
(choice) {                case 1:
                System.out.print("Enter account number: ");
int accNum = scanner.nextInt();
                System.out.print("Enter deposit amount: INR ");
double depositAmount = scanner.nextDouble();
for (Account acc : accounts) {                    if
(acc.accountNumber == accNum) {
acc.deposit(depositAmount);
                    }
}
break;
case 2:
                System.out.print("Enter account number: ");
accNum = scanner.nextInt();
```

```
                System.out.print("Enter withdrawal amount: INR ");
double withdrawAmount = scanner.nextDouble();
for (Account acc : accounts) {                    if
(acc.accountNumber == accNum) {
acc.withdraw(withdrawAmount);
                }
}
break;
case 3:
                System.out.print("Enter account number: ");
accNum = scanner.nextInt();                    for (Account
acc : accounts) {                    if (acc.accountNumber
== accNum) {                    acc.displayBalance();
                }
}
break;
case 4:
                System.out.print("Enter account number (for Savings account): ");
accNum = scanner.nextInt();                    for (Account acc : accounts) {
if (acc.accountNumber == accNum && acc instanceof SavAcct) {
                    ((SavAcct) acc).computeInterest();
                }
}                break;
case 5:
exit = true;
break;
default:
                System.out.println("Invalid choice. Please enter a valid option.");
            }
        }
```

```
    }
}
```

Output:

```
C:\Users\bmsce\Desktop>java Bank

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit
Enter your choice: 1
Enter amount to deposit: 10000
Select account (1. Current, 2. Savings): 2
Deposit of $10000.0 successful. Updated balance: $12000.0

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit
Enter your choice: 2
Select account (1. Current, 2. Savings): 1
Account Balance: $1000.0

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit
Enter your choice: 3
Interest computed and deposited: $600.0
Account Balance: $12600.0

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit
Enter your choice: 4
Enter amount to withdraw: 3000
Select account (1. Current, 2. Savings): 1
Insufficient funds. Withdrawal failed.

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit
Enter your choice: 5
```

Lab 6

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
package CIE; public class Student {     public String
usn;     public String name;     public int sem;
public Student(String usn, String name, int sem) {
this.usn = usn;         this.name = name;         this.sem
= sem;
    }
}
package CIE;
public class Internals extends Student {     public int[] internalMarks;
public Internals(String usn, String name, int sem, int[] internalMarks) {
super(usn, name, sem);         this.internalMarks = internalMarks;
    }
}
```

```
package SEE;
import CIE.Student;
public class External extends Student {     public int[] seeMarks;
public External(String usn, String name, int sem, int[] seeMarks) {
super(usn, name, sem);         this.seeMarks = seeMarks;
```

```
    }
}

import CIE.Internals; import SEE.External; import
java.util.Scanner; public class FinalMarks {     public
static void main(String[] args) {         Scanner scanner =
new Scanner(System.in);         System.out.print("Enter
the number of students: ");         int n =
scanner.nextInt();
        Internals[] cieStudents = new Internals[n];
External[] seeStudents = new External[n];         for
(int i = 0; i < n; i++) {
            System.out.println("Enter details for CIE of student " + (i + 1));
            System.out.print("USN: ");
            String usn = scanner.next();
            System.out.print("Name: ");
            String name = scanner.next();
System.out.print("Semester: ");
int sem = scanner.nextInt();
int[] cieMarks = new int[5];
            System.out.print("Enter CIE marks for 5 courses: ");
for (int j = 0; j < 5; j++) {
                cieMarks[j] = scanner.nextInt();
            }
            cieStudents[i] = new Internals(usn, name, sem, cieMarks);
        }
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for SEE of student " + (i + 1));
            System.out.print("USN: ");
            String usn = scanner.next();
```

```
            System.out.print("Name: ");
            String name = scanner.next();
System.out.print("Semester: ");
int sem = scanner.nextInt();            int[]
seeMarks = new int[5];
            System.out.print("Enter SEE marks for 5 courses: ");
for (int j = 0; j < 5; j++) {                seeMarks[j] =
scanner.nextInt();
             }
            seeStudents[i] = new External(usn, name, sem, seeMarks);
        }
        System.out.println("\nFinal Marks of Students:");
for (int i = 0; i < n; i++) {
            System.out.println("\nDetails of Student " + (i + 1));
            System.out.println("USN: " + cieStudents[i].usn);
            System.out.println("Name: " + cieStudents[i].name);
            System.out.println("Semester: " + cieStudents[i].sem);
System.out.println("CIE Marks: ");            for (int j = 0; j <
5; j++) {
                System.out.print(cieStudents[i].internalMarks[j] + " ");
            }
            System.out.println("\nSEE Marks: ");
            for (int j = 0; j < 5; j++) {
                System.out.print(seeStudents[i].seeMarks[j] + " ");
            }
        }
    }
}
```

Output:

```
PS C:\Users\Admin\Desktop\1bm22cs187> javac Final.java
PS C:\Users\Admin\Desktop\1bm22cs187> java Final
Enter the value of n
2
Enter the usn:
1
Enter the Name:
A
Enter the Sem:
1
Enter marks in 5 subjects:
Marks in :1subject is
45
Marks in :2subject is
34
Marks in :3subject is
50
Marks in :4subject is
44
Marks in :5subject is
46
Enter marks in 5 subjects:
Marks in :1subject is
33
Marks in :2subject is
45
Marks in :3subject is
50
Marks in :4subject is
44
Marks in :5subject is
45
Name is:A
USN is:1
Sem is:1
Internal details
Marks in :1subject is45
Marks in :2subject is34
Marks in :3subject is50
Marks in :4subject is44
Marks in :5subject is46
External details
Marks in :1subject is33
Marks in :2subject is45
Marks in :3subject is50
Marks in :4subject is44
```

```
Marks in :5subject is45
Marks in :1subject is78
Marks in :1subject is79
Marks in :1subject is100
Marks in :1subject is88
Marks in :1subject is91
Enter the usn:
2
Enter the Name:
        B
Enter the Sem:
1
Enter marks in 5 subjects:
Marks in :1subject is
45
Marks in :2subject is
44
Marks in :3subject is
35
Marks in :4subject is
50
Marks in :5subject is
32
Enter marks in 5 subjects:
Marks in :1subject is
45
Marks in :2subject is
50
Marks in :3subject is
45
Marks in :4subject is
29
Marks in :5subject is
44
Name is:B
USN is:2
Sem is:1
Internal details
Marks in :1subject is45
Marks in :2subject is44
Marks in :3subject is35
Marks in :4subject is50
Marks in :5subject is32
External details
Marks in :1subject is45
Marks in :2subject is50
Marks in :3subject is45
Marks in :4subject is29
Marks in :5subject is44
Marks in :2subject is90
```

```
External details
Marks in :1subject is45
Marks in :2subject is50
Marks in :3subject is45
Marks in :4subject is29
Marks in :5subject is44
Marks in :2subject is90
Marks in :2subject is94
Marks in :2subject is80
Marks in :2subject is79
Marks in :2subject is76
PS C:\Users\Admin\Desktop\1bm22cs187>
```

Lab 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

```
import java.util.Scanner; class
WrongAge extends Exception {
public WrongAge(String message) {
super(message);
   }
} class Father {     protected int fatherAge;     public Father(int
age) throws WrongAge {         fatherAge = age;         if
(fatherAge < 0) {             throw new WrongAge("Father's age
cannot be negative");
       }
   }
```

```
} class Son extends Father {     private int sonAge;     public Son(int
fatherAge, int sonAge) throws WrongAge {         super(fatherAge);
this.sonAge = sonAge;         if (sonAge <= 0) {             throw new
WrongAge("Son's age cannot be negative or zero");
        }
        if (sonAge >= fatherAge) {             throw new WrongAge("Son's age cannot be
greater than or equal to father's age");
        }
    }
} public class Main
{
    public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
try {
            System.out.print("Enter father's age: ");
int fatherAge = scanner.nextInt();
System.out.print("Enter son's age: ");             int
sonAge = scanner.nextInt();             Son son =
new Son(fatherAge, sonAge);
            System.out.println("Father's age: " + fatherAge);
            System.out.println("Son's age: " + sonAge);
        } catch (WrongAge e) {
            System.out.println("Exception caught: " + e);
            System.out.println("Exception caught: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("Error: " + e);
            System.out.println("Error: " + e.getMessage());
        } finally {
scanner.close();
        }
```

```
    }
}
```

Output:

```
Enter father's age: 50
Enter son's age: -20
Exception caught: WrongAge: Son's age cannot be negative or zero
Exception caught: Son's age cannot be negative or zero
```

Lab 8

Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```
class DisplayThread extends Thread {     private String
message;     private int interval;     private boolean
running = true;     public DisplayThread(String
message, int interval) {         this.message = message;
this.interval = interval;
    }
    public void run() {
while (running) {
            System.out.println(message);
try {
                Thread.sleep(interval);
            } catch (InterruptedException e) {
e.printStackTrace();
            }
        }
```

```
    }
    public void stopThread() {
running = false;
    }
} public class ThreadEx
{
    public static void main(String[] args) {
        DisplayThread bmsThread = new DisplayThread("BMS College of Engineering", 10000);
DisplayThread cseThread = new DisplayThread("CSE", 2000);        bmsThread.start();
cseThread.start();
        System.out.println("Press Enter to stop the threads...");
try {
            System.in.read();
} catch (Exception e) {
e.printStackTrace();
        }
        bmsThread.stopThread();
cseThread.stopThread();
    }
}
```

Output:

```
PS C:\Users\Admin\Desktop\1BM22CS187> javac Thread1.java
PS C:\Users\Admin\Desktop\1BM22CS187> java Thread1
Press Enter to stop the threads...
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```

Lab 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

```
import javax.swing.*;
import java.awt.*; import
```

```
java.awt.event.*; class
SwingDemo{
SwingDemo(){
JFrame jfrm = new JFrame("Divider App");
jfrm.setSize(275, 150); jfrm.setLayout(new
FlowLayout());
jfrm.setDefaultCloseOperation(JFrame.EXI
T_ON_CLOSE);
JLabel jlab = new JLabel("Enter the divider and divident:");
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
JButton button = new JButton("Calculate");
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel(); JLabel
anslab = new JLabel();
jfrm.add(err); // to display error
bois jfrm.add(jlab); jfrm.add(ajtf);
jfrm.add(bjtf); jfrm.add(button);
jfrm.add(alab); jfrm.add(blab);
jfrm.add(anslab);
ActionListener l = new ActionListener() { public
void actionPerformed(ActionEvent evt) {
System.out.println("Action event from a text
field");
} }; ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener()
{ public void actionPerformed(ActionEvent evt)
```

```
{ try{ int a = Integer.parseInt(ajtf.getText()); int
b = Integer.parseInt(bjtf.getText()); int ans = a/b;
alab.setText("\nA = " + a); blab.setText("\nB
= " + b); anslab.setText("\nAns = "+ ans);
} catch(NumberFormatException
e){ alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("Enter Only Integers!");
} catch(ArithmeticException e){
alab.setText(""); blab.setText("");
anslab.setText(""); err.setText("B
should be NON zero!");
}
} });
jfrm.setVisible(true);
}
public static void main(String args[]){
SwingUtilities.invokeLater(new
Runnable(){ public void run(){ new
SwingDemo();
}
});
}
}
```

Output:

Lab10

Demonstrate Inter process Communication and deadlock. IPC

```
class Q { int
n;
boolean valueSet = false;
synchronized int get() {
while(!valueSet) try {
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
System.out.println("Got: " + n);
valueSet = false; notify();
return n; } synchronized void
put(int n) { while(valueSet)
try { wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
} this.n = n; valueSet = true;
System.out.println("Put: " + n);
notify(); }
}
class Producer implements Runnable {
```

```
Q q; Producer(Q
q) { this.q = q;
new Thread(this, "Producer").start();
} public void run()
{ int i = 0;
while(i<15) {
q.put(i++);
}
}
}
class Consumer implements Runnable {
Q q; Consumer(Q
q) {
this.q = q; new Thread(this,
"Consumer").start();
} public void run()
{ int i=0;
while(i<15) { int
r=q.get(); i++; }
} } class PCFixed { public static void
main(String args[]) { Q q = new Q();
new Producer(q); new Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
```

Output:

```
Press Control-C to stop.
Put: 0
Got: 0
Put: 1
Got: 1
Put: 2
Got: 2
Put: 3
Got: 3
Put: 4
Got: 4
Put: 5
Got: 5
Put: 6
Got: 6
Put: 7
Got: 7
Put: 8
Got: 8
Put: 9
Got: 9
Put: 10
Got: 10
Put: 11
Got: 11
Put: 12
Got: 12
Put: 13
Got: 13
Put: 14
Got: 14
```

Deadlock

class A { synchronized void

foo(B b) {

```
String name = Thread.currentThread().getName();
System.out.println(name + " entered A.foo"); try
{
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("A Interrupted");
}
System.out.println(name + " trying to call B.last()");
b.last(); } void last() {
System.out.println("Inside A.last");
} } class B { synchronized
void bar(A a) {
String name = Thread.currentThread().getName();
System.out.println(name + " entered B.bar"); try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("B Interrupted");
}
System.out.println(name + " trying to call A.last()");
a.last(); } void last() {
System.out.println("Inside A.last");
}
} class Deadlock implements
Runnable
{
A a = new A();
B b = new B();
Deadlock() {
Thread.currentThread().setName("MainThread"); Thread t = new
Thread(this,"RacingThread");
t.start();
```

```
a.foo(b);
System.out.println("Back in mainthread");
} public void run()
{ b.bar(a);
System.out.println("Back in other thread");
} public static void main(String args[])
{ new Deadlock();
}
}
```

Output:

```
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
Inside A.last
Back in main thread
RacingThread trying to call A.last()
Inside A.last
Back in other thread
```